T-1.55-RT

AD 710227

A COMPARISON OF TWO QUADRATIC
PROGRAMMING ALGORITHMS

Technical Report No. 41

PROJECT THEMIS

Systems Research Center
Industrial & Systems Engineering Department
University of Florida
Gainesville

21

A COMPARISON OF TWO QUADRATIC
PROGRAMMING ALGORITHMS

Technical Report No. 41

SUBMITTED BY:

Gerald L. Hefley
Research Associate

M. E. Thomas
Principal Investigator

APPROVED BY:

B. D. Sivazlian
Director, Project THEMIS

Department of Industrial and Systems Engineering
The University of Florida
Gainesville, Florida 32601

June 1970

This research was performed under Project THEMIS, ARO-D Contract No. DAH C04 68C0002.

ABSTRACT

This paper compares Wolfe's quadratic programming algorithm with Cottle and Dantzig's principle pivot method. It is shown that Wolfe's algorithm requires more operations.

TABLE OF CONTENTS

In this paper we will compare Wolfe's quadratic programming algorithm with the principle pivot method proposed by Cottle and Dantzig [3]. In the first section we describe the problem and define some notation. In sections 2 and 3 the algorithms are briefly described and examples worked out. Then in section 4 we discuss the merits of each algorithm.

No typographical distinction is made between vectors and scalars. Matrices are designated by capital letters. The transpose of w is written w'. All vectors are column vectors.

1. <u>Problem Formulation</u> The problem we wish to consider may be formulated as follows. Find $\hat{x}$ such that

$$f(\hat{x}) = \min_{x \varepsilon R}(c'x + \frac{1}{2}x'Dx) \tag{1.1}$$

where $R = \{x \varepsilon E^n;\ Ax \geqq b, \quad x \geqq 0\}$,

$c \varepsilon E^n$, $b \varepsilon E^m$

A is a mxn matrix and D is a nxn matrix.

We can assume without loss of generality that D is a symmetric matrix.

If the region R is empty, then the problem (1.1) has no solution. If R contains a single point, then the problem has a trivial, unique solution. We will, therefore, assume that the interior of the region R is non-empty.

Define the Lagrange function

$$L(x, y, v) = c'x + \frac{1}{2}x'Dx - y'(Ax - b) - v'x \tag{1.2}$$

where $y \varepsilon E^m$, $v \varepsilon E^n$.

Under the assumptions made on R we know that the Kuhn-Tucker [5] conditions

are necessary. That is, $\hat{x}$ exist only if there exist vectors $\hat{y}$ and $\hat{v}$ such that $\hat{x}$, $\hat{y}$, $\hat{v}$ satisfy

$$\left.\begin{aligned} & c' + x'D - y'A - v' = 0 \\ & Ax - b \geq 0 \\ & y'(Ax - b) = 0, \ v'x = 0 \\ & x \geq 0, \ y \geq 0, \ v \geq 0. \end{aligned}\right\} \quad (1.3)$$

Solving these equations, eliminating v, rearranging and recalling that $D' = D$ we have

$$\left.\begin{aligned} & c + Dx - A'y \geq 0 \\ & -b + Ax \geq 0 \\ & y'(-b + Ax) = 0, \quad x'(c + Dx - A'y) = 0 \\ & x \geq 0, \quad y \geq 0. \end{aligned}\right\} \quad (1.4)$$

We can express these conditions in a more compact way. Put

$$z = \begin{bmatrix} x \\ y \end{bmatrix}, \quad q = \begin{bmatrix} c \\ -b \end{bmatrix}, \quad M = \begin{bmatrix} D & -A' \\ A & 0 \end{bmatrix}. \quad (1.5)$$

Then (1.4) is equivalent to what Cottle and Dantzig [3] call the Fundamental Problem.

$$\left.\begin{aligned} & w = q + Mz \\ & w'z = 0 \\ & w \geq 0, \quad z \geq 0 \end{aligned}\right\} \quad (1.6)$$

where M is a $(n+m)x(n+m)$ matrix,

and $w, q, z \in E^{n+m}$.

We shall be concerned with two different algorithms which can be used to solve the Kuhn-Tucker conditions as expressed in (1.6).

2. Wolfe's Algorithm The set of equations in (1.6) is linear except for $w'z = 0$. Wolfe's algorithm is a modification of linear programming algorithms to account for the equation $w'z = 0$.

Make $b \geqq 0$ in (1.4) and subtract and add the slack and surplus variables t_j such that $t \geqq 0$, $t \varepsilon E^m$. Let G be the matrix $[\pm\delta_{ij}]$ of coefficients of t. Now define the surplus vector $s \geqq 0$, $s \varepsilon E^n$ and rewrite (1.4) as

$$\left.\begin{aligned} & c + Dx - A'y - s = 0 \\ & -b + Ax - Gt = 0 \\ & y'(-b + Ax) = 0, \quad x'(c + Dx - A'y) = 0 \\ & x, s \geqq 0 \quad y, t \geqq 0 . \end{aligned}\right\} \quad (2.1)$$

Or equivalently,

$$\left.\begin{aligned} & \begin{bmatrix} c \\ -b \end{bmatrix} + \begin{bmatrix} D & -A' & -I & 0 \\ A & 0 & 0 & -I \end{bmatrix} \begin{bmatrix} x \\ y \\ s \\ Gt \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ & (x', y') \begin{bmatrix} s \\ Gt \end{bmatrix} = 0 \\ & [x, y, s, t] \geqq 0 . \end{aligned}\right\} \quad (2.2)$$

Now using the definitions in (1.5) and putting $w = [s, Gt]$ we have

$$\left.\begin{aligned} & q = (M, -I) \begin{bmatrix} z \\ w \end{bmatrix} = 0 \\ & w'z = 0 \\ & w \geqq 0, \quad z \geqq 0 \end{aligned}\right\} \quad (2.3)$$

which is of course (1.6) reformatted.

Now consider the problem

$$\left.\begin{aligned}&\min \sum_{i=1}^{n} u_i\\&\text{subject to}\\&(M,\ -\ I,\ I)\begin{bmatrix} z \\ w \\ u \end{bmatrix} = -\ q\\&w \geqq 0, \quad z \geqq 0, \quad u \geqq 0.\end{aligned}\right\} \quad (2.4)$$

If the solution to (2.4) has $u = 0$ and at the same time we have $w'z = 0$, then we have solved (1.4) and hence (1.1) our original problem.

Wolfe's algorithm is to solve (2.4) using a linear programming algorithm modified to ensure that $w'z = 0$ at all times. We can hold $w'z = 0$ if at each iteration we ensure that if one of the vectors associated with w_k or z_k is in the basis then the other is not.

The condition $w'z = 0$ guarantees that no more than (n+m) variables in (2.3) need be non-zero. Hence, the solution to (2.3) is one of the basic solutions to (2.4). In fact it can be shown that in spite of the requirement that $w'z = 0$, the linear programming algorithms still converge. (For example, see Hadley [4] pages 214-221.)

Wolfe's algorithm requires that every iteration maintain $w'z = 0$, hence we must start with a basic feasible solution for which this is true. Since q has (n+m) components, if we put

$$z^o = 0$$

$$s_i^o = \begin{cases} c_i & c \geqq 0 \\ 0 & \text{otherwise,} \end{cases} \qquad i = 1, \ldots, n$$

$$u_i^o = \begin{cases} -c_i & c < 0 \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, \ldots, n$$

and

$$Gt^o = -b.$$

Then $[z^o, w^o, u^o]$ is an initial basic feasible solution to (2.4) which satisfies (2.3).

Convergence is guaranteed (assuming no degeneracy) if D is positive definite. Hadley [4] describes a method due to Charnes [1] for solving the semidefinite case. Instead of using D we use $(D + \epsilon I)$ for $\epsilon > 0$ and arbitrarily small. Then $(D + \epsilon I)$ is positive definite and we have convergence.

We will now consider two examples. The first is positive definite and the second is positive semidefinite. However, both problems have unique solutions.

The first problem we wish to consider is

$$\min x_1^2 + x_2^2 - 8x_1 - 10x_2,$$

subject to

$$3x_1 + 2x_2 \leq 6,$$

$$x_1, x_2 \geq 0.$$

For this problem we have

$$c = \begin{bmatrix} -8 \\ -10 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad A = (-3, -2), \quad b = (-6).$$

And Wolfe's algorithm gives us (from 2.4)

$$\min u_1 + u_2$$

subject to

$$\begin{bmatrix} 2 & 0 & 3 & -1 & 0 & 0 & 1 & 0 \\ 0 & 2 & 2 & 0 & -1 & 0 & 0 & 1 \\ -3 & -2 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ w \\ u \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \\ -6 \end{bmatrix}$$

with an initial basic feasible solution

$[z^o, w^o, u^o] = [0, 0, 0, 0, 0, 6, 8, 10]$.

Tableaux I through IV are for the above problem. We see in tableaux I and III that we cannot apply the normal simplex rules for exchanges of basis vectors and Wolfe's modification comes into play. The optimal solution to this problem is $[\hat{z}, \hat{w}, \hat{u}]=[4/13, 33/13, 32/13, 0, 0, 0, 0, 0]$.

TABLEAU I: INITIAL TABLEAU

z_1	z_2	z_3	w_1	w_2	w_3	u_1	u_2		
2	0	3	-1	0	0	1	0	8	u_1
0	2	2	0	-1	0	0	1	10	u_2
-3	-2	0	0	0	-0	0	0	-6	w_3
2	2	5	-1	-1	0	0	0	18	

TABLEAU II

0	-4/3	3	-1	0	-2/3	1	0	4	u_1
0	2	2	0	-1	0	0	1	10	u_2
1	2/3	0	0	0	1/3	0	0	2	z_1
0	2/3	5	-1	-1	-2/3	0	0	14	

TABLEAU III

0	-4/9	1	-1/3	0	-2/9	1/3	0	4/3	z_3
0	26/9	0	2/3	-1	4/9	-2/3	1	22/3	u_2
1	2/3	0	0	0	1/3	0	0	2	z_1
0	26/9	0	2/3	-1	4/9	-5/3	0	22/3	

TABLEAU IV

0	0	1	-3/13	-2/13	-2/13	3/13	2/13	32/13	z_3
0	1	0	3/13	-9/26	2/13	-3/13	9/26	33/13	z_2
1	0	0	-2/13	-3/13	3/13	2/13	-3/13	4/13	z_1
0	0	0	0	0	0	-1	-1	0	

Our second example is

$$\min\ 2x_2^2 - 2x_1 - 3x_2$$

subject to

$$x_1 + 4x_2 \leqq 4$$

$$x_1 + x_2 \leqq 2 .$$

For this problem we have

$$c = \begin{bmatrix} -2 \\ -3 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & -4 \\ -1 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} -4 \\ -2 \end{bmatrix}.$$

Clearly, D is semidefinite. Our initial basic feasible solution is $[z^o, w^o, u^o] = [0, 0, 0, 0, 0, 0, 4, 2, 2, 3]$. And the optimal solution is

$$[\hat{z}, \hat{w}, \hat{u}] = [7/4, 1/4, 0, 2, 0, 0, 5/4, 0, 0, 0].$$

This problem takes five iterations (Tableaux V thru X). Note that in Tableau VI we cannot enter z_3 or z_4 into the basis so that z_1 must enter the basis even though the last row is 0.

TABLEAU V

z_1	z_2	z_3	z_4	w_1	w_2	w_3	w_4	u_1	u_2		
0	0	1	1	-1	0	0	0	1	0	2	u_1
0	4	4	1	0	-1	0	0	0	1	3	u_2
-1	-4	0	0	0	0	-1	0	0	0	-4	w_3
-1	-1	0	0	0	0	0	-1	0	0	-2	w_4
0	4	5	2	-1	-1	0	0	0	0	5	

TABLEAU VI

0	0	1	1	-1	0	0	0	1	0	2	u_1
0	1	1	1/4	0	-1/4	0	0	0	1/4	3/4	z_2
-1	0	4	1	0	-1	-1	0	0	1	-1	w_3
-1	0	1	1/4	0	-1/4	0	-1	0	1/4	-5/4	w_4
0	0	1	1	-1	0	0	0	0	-1	2	

TABLEAU VII

0	0	1	1	-1	0	0	0	1	0	2	u_1
0	1	1	1/4	0	-1/4	0	0	0	1/4	3/4	z_2
1	0	-4	-1	0	1	1	0	0	-1	1	z_1
0	0	-3	-3/4	0	3/4	1	-1	0	-3/4	-1/4	w_4
0	0	1	1	-1	0	0	0	0	-1	2	

TABLEAU VIII

0	0	0	3/4	-1	1/4	1/3	-1/3	1	-1/4	23/12	u_1
0	1	0	0	0	0	1/3	-1/3	0	0	8/12	z_2
1	0	0	0	0	0	-1/3	4/3	0	0	4/3	z_1
0	0	1	1/4	0	-1/4	-1/3	1/3	0	1/4	1/12	z_3
0	0	0	3/4	-1	1/4	1/3	-1/3	0	-1/4	23/12	

TABLEAU IX

z_1	z_2	z_3	z_4	w_1	w_2	w_3	w_4	u_1	u_2		
0	0	-3	0	-1	1	4/3	-4/3	1	-1	5/3	u_1
0	1	0	0	0	0	1/3	-1/3	0	0	8/12	z_2
1	0	0	0	0	0	-1/3	4/3	0	0	4/3	z_1
0	0	4	1	0	-1	-4/3	4/3	0	1	1/3	z_4
0	0	-3	0	-1	1	4/3	-4/3	0	-1	5/3	

TABLEAU X

0	0	-9/4	0	-3/4	3/4	1	-1	3/4	-3/4	5/4	w_3
0	1	3/4	0	1/4	-1/4	0	0	-1/4	1/4	1/4	z_2
1	0	-3/4	0	-1/4	1/4	0	1	1/4	-1/4	7/4	z_1
0	0	1	1	-1	0	0	0	1	0	2	z_4
0	0	0	0	0	0	0	0	-1	-1	0	

3. The Principle Pivot Method For convenience we repeat the Fundamental Problem. Find vectors w, z, such that

$$\left.\begin{aligned} & w = q + Mz, \\ & w'z = 0, \\ & z \geq 0,\ w \geq 0 \end{aligned}\right\} \quad (3.1)$$

where M is a (n + m) square matrix and $w, q, z \in E^{n+m}$. We have seen that this system (3.1) is equivalent to the Kuhn-Tucker conditions (1.4) and has a unique solution if D is positive definite.

From (3.1) we have $w'z = \Sigma_{i=1}^{n+m} w_i z_i = 0$ which implies that each non-negative pair (w_i, z_i) has at least one zero component. We call (w_i, z_i) the complementary pair.

A pivot is an operation on the equation w = q + Mz which exchanges the roles of w_j and z_k for some j and k. We call (w_j, z_k) the pivotal pair. The pivot is accomplished by solving one of the (n+m) equations for z_k in terms of z_i, $i \neq k$ and w_j. We then eliminate the variable z_k from the other (n+m) - 1 equations. The result is a new system

$$w^v = q^v + M^v z^v \quad (3.2)$$

in which the roles of the pivotal pair are interchanged, i.e., w_j appears on the right and z_k appears on the left in (3.2). The operation is not defined if $m_{jk} = 0$. If in the pivotal pair, j = k, then it is called a complementary pivot. A sequence of complementary pivots is a principle pivot. The reader will recognize the pivot operation as an iteration of the classic Gauss-Jordan reduction technique for solving linear equations.

A basic solution to (3.2) is a solution with no more than (n+m) non-zero variables. A basic solution to (3.2) which also satisfies (3.1) is

a complementary basic solution.

Cottle and Dantzig [3] show that, under certain assumptions on the matrix M, a sequence of principle pivots on (3.2) starting from a basic solution to (3.2) will converge to a complementary basic solution to (3.1) in a finite number of iterations (assuming no degeneracy). This is the Principle Pivot Method. Convergence of this algorithm is proved in the case where M has positive principle minors [3]. If M is positive definite, then it has positive principle minors. In the case where M is positive semidefinite, some of the principle minors are zero. Cottle [2] has provided a modification of the principle pivot algorithm which converges if M is positive semidefinite to a solution or an indication of an unbounded solution. The proofs for convergence depend on the observations that the properties of "positive principle minors" and "positive (semi)definite" remain invariant under the operation of principle pivot.

To translate the requirements on M to equivalent requirements on D we observe that if D is positive definite or positive semidefinite, then M is positive semidefinite since

$$\begin{aligned} z'Mz &= (x', y') \begin{bmatrix} D & -A' \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ &= x'Dx + (y'Ax - x'A'y) \\ &= x'Dx \,. \end{aligned}$$

Hence, if D is positive (semi)definite we know that the algorithm will converge or give an indication of an unbounded solution.

We now consider the examples solved by Wolfe's method in section 2. We will use the tableau suggested by Cottle [2]. The first problem is in tableaux XI through XIV and the second problem in tableaux XV through XVIII.

TABLEAU XI

		z_1	z_2	z_3
w_1	-8	2	0	3
w_2	-10	0	2	2
w_3	6	-3	-2	0

TABLEAU XII

		w_3	z_2	z_3
w_1	-4	0	-4/3	3
w_2	-10	0	2	2
z_1	2	-0	-2/3	0

TABLEAU XIII

		w_3	z_2	w_1
z_3	4/3	0	4/9	-1
w_2	-22/3	0	26/9	0
z_1	2	-1	-2/3	0

TABLEAU XIV

		w_3	w_2	w_1
z_3	32/13	0	0	-1
z_2	33/13	0	-1	0
z_1	4/13	-1	0	0

TABLEAU XV

		z_1	z_2	z_3	z_4
w_1	-2	0	0	1	1
w_2	-3	0	4	4	1
w_3	4	-1	-4	0	0
w_4	2	-1	-1	0	0

TABLEAU XVI

		w_4	z_2	z_3	z_4
w_1	-2	0	0	1	1
w_2	-3	0	4	4	1
w_3	2	0	-3	0	0
z_1	2	-1	-1	0	0

TABLEAU XVII

		w_4	z_2	z_3	w_1
z_4	2	0	0	-1	-1
w_2	-1	0	4	3	0
w_3	2	0	-3	0	0
z_1	2	-1	-1	0	0

TABLEAU XVIII

		w_4	w_2	z_3	w_1
z_4	2	0	0	-1	-1
z_2	1/4	0	-1	-3/4	0
w_3	5/4	0	0	4/4	0
z_1	7/4	-1	0	3/4	0

4. Comparison Both of the algorithms described in sections 2 and 3 are adjacent extreme point methods. That is, each algorithm starts at an extreme point of a closed convex set and proceeds systematically by an exchange of basis vectors to examine an adjacent extreme point.

If we use the simplex algorithm in Wolfe's technique then we see that each iteration involves a pivot operation. However, the matrix on which this operation is performed is at least $(n+m)x2(n+m)$[1] in the case of Wolfe's algorithm and $(n+m)x(n+m)$ in the principle pivot algorithm. This means four times, or more, as many operations for Wolfe's method per iteration and at least twice the storage requirement. The revised simplex method offers some improvement in Wolfe's algorithm but it is not significant enough to balance the scales in the direction of Wolfe's method.

We have already observed that both algorithms converge under the same circumstances. This is due primarily to the use of the Kuhn-Tucker conditions in establishing the problem. However, the semidefinite case may have an unbounded solution. The principle pivot method with Cottle's modification detects this situation. Wolfe's algorithm with Charnes' modification, however changes the problem itself. It will converge even though the solution is unbounded. As an example, consider

$$\min \quad x_1^2 - 2x_1 - 3x_2,$$

subject to

$$x \geqq 0.$$

This problem has an unbounded solution at $(1, x_2)$. However, by adding $\varepsilon \geqq 0$ we have

---

1 This number is increased by n using the method described in this paper for choosing an initial basic feasible solution. Hadley [4] describes an alternative method which requires only 2(n+m) columns. However, it involves finding the inverse of the basis matrix.

$$\min\ (1 + \frac{1}{2}\,\varepsilon)\ x_1^2 + \frac{1}{2}\,\varepsilon x_2^2 - 2x_1 - 3x_2,$$

subject to $x \geqq 0$.

Which has a solution at $(1/(1 + \frac{1}{2}\,\varepsilon),\ 3/\varepsilon)$.

Clearly, as $\varepsilon \to 0$ we see that the solution is unbounded. The problem of course is that we may not recognize it as unbounded in practice if ε is too large. (Because of the capacity of a computer for example.) With care this disadvantage can be overlooked.

One interesting difference between the two methods is that in Wolfe's method the solution at each iteration is feasible. However, in the principle pivot method the solution is not feasible until convergence. This is not very important since it would be useful only if one stopped the iterations before optimality was reached. Because of the relative speeds, this is more likely to happen with Wolfe's algorithm anyway.

We have seen that Wolfe's method requires at least four times as many operations per iteration as the principle pivot method. The principle pivot method should require no more than (n+m) iterations while Wolfe's will generally require from (n+m) to 2(n+m) iterations. From this observation, one easily concludes that the principle pivot method is faster than Wolfe's and hence superior since it solves the same class of problems as Wolfe's.

BIBLIOGRAPHY

1. Charnes, A., and W. W. Cooper, Management Models and Industrial Applications of Linear Programming, Vol. II, Wiley, New York, 1961, pp. 682-687.
2. Cottle, Richard W., "The Principle Pivoting Method of Quadratic Programming," Mathematics of the Decision Sciences, Part I, G. B. Dantzig and A. F. Veinott, Jr., ed., American Mathematical Society, Providence, R. I., 1968.
3. Cottle, Richard W., and George B. Dantzig, "Complementary Pivot Theory of Mathematical Programming," Mathematics of the Decision Sciences, Part 1, G. B. Dantzig and A. F. Veinott, Jr., ed., American Mathematical Society, Providence, R. I., 1968.
4. Hadley, G., Nonlinear and Dynamic Programming, Addison-Wesley, Reading, Mass., 1964.
5. Kuhn, H. W., and A. W. Tucker, "Nonlinear Programming," Second Berkeley Symposium on Mathematical Statistics and Probability, Neyman, J., ed., University of California Press, Berkeley, California, 1951.
6. Wolfe, P., "The Simplex Method for Quadratic Programming," Econometrica, Vol. 27, 1959, pp. 382-398.

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of Florida	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP Unclassified

3. REPORT TITLE

"A Comparison of Two Quadratic Programming Algorithms"

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (Last name, first name, initial)

Hefley, Gerald L.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1970	16	6

8a. CONTRACT OR GRANT NO. DAH C04 68C0002 b. PROJECT NO. 2T0 14501B81B c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Report No. 41 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. AVAILABILITY/LIMITATION NOTICES

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	ARO-Durham

13. ABSTRACT

This paper compares Wolfe's quadratic programming algorithm with Cottle and Dantzig's principle pivot method. It is shown that Wolfe's algorithm requires more operations.